Computational Neuroscience

Saccade detection using a particle filter

Pierre M. Daye a,*,b,†, Lance M. Optican b

a ICM, Pierre et Marie Curie Paris-6 Université, INSERM UMRS 975, CNRS 7225, F-75013, Paris, France
b Laboratory of Sensorimotor Research, National Eye Institute, National Institutes of Health, Bethesda, MD, USA

HIGHLIGHTS

• Detect saccades using a particle filter as Bayesian estimator.
• Remove the baseline velocity signal linked to smooth velocity or eye drifts, allowing a constant threshold for the detection.
• Validated with five different paradigms, microsaccades, microsaccades plus saccades with drift, linear target motion, non-linear target motion and free viewing.
• Noise sensitivity analysis.

ARTICLE INFO

Article history:
Received 18 March 2014
Received in revised form 27 May 2014
Accepted 16 June 2014
Available online 17 July 2014

Keywords:
Catch-up saccade
Smooth pursuit
Eye drift
Microsaccades
Bayesian estimator

ABSTRACT

Background: When healthy subjects track a moving target, "catch-up" saccades are triggered to compensate for the non-perfect tracking gain. The evaluation of the pursuit and/or saccade kinematics requires that saccade and pursuit components be separated from the eye movement trace. A similar situation occurs when analyzes eye movements of patients that could contain eye drifts between saccades. This task is especially difficult, because the range of saccadic amplitudes goes from microsaccades (less than 1°) to large exploratory saccades (40°).

New method: In this paper we proposed a new algorithm to detect saccades based on a particle filter. The new method suppresses the baseline velocity component linked to smooth pursuit (or to eye drifts) and thus permits a constant threshold during a trial despite the smooth pursuit behavior. It also accounts for a wide range of saccade amplitudes.

Results: The new method is validated with five different paradigms, microsaccades, microsaccades plus saccades with drift, linear target motion, non-linear target motion and free viewing. The sensitivity of the method to signal noise is analyzed.

Comparison with existing methods: Traditional saccade detection algorithms using a velocity (or acceleration or jerk) threshold can be inadequate because of the baseline velocity component linked to the smooth pursuit (especially if the target motion is non-linear, i.e. not constant velocity) or to eye drifts between saccades.

Conclusions: The new method detects saccades in challenging situations involving eye movements between saccades (smooth pursuit and/or eye drifts) and unfiltered recordings.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Foveate vertebrates orient their gaze using mainly three types of movements that have been classified since the early 1900s (Dodge, 1903). High speed eye movements called saccades shift gaze from one center of interest to another. Subjects track a moving target using smooth pursuit and finally they use vergence movements to align both eyes on the same target. Eye movement studies require the ability to identify accurately which parts of the movement correspond to fixations (stable gaze), saccades (fast redirection of the line of sight) or smooth pursuit (tracking of a moving target).

Once an eye trace has been segmented into its basic components, one can extract parameters that will characterize the subtype of movement studied. A classical example used to characterize saccades is the main sequence, which links saccade amplitude to...
saccade peak velocity or to saccade duration (Bahill et al., 1975). The main sequence has proven to be a good diagnostic tool to detect abnormal saccadic eye movements (e.g. Zee et al., 1976; Daye et al., 2013). The analysis of smooth pursuit can be more tricky because when a subject tracks a moving target, the pursuit gain (ratio between eye velocity and target velocity) is generally smaller than one (Dodge, 1903; Meyer et al., 1985). Therefore, a positional error arises and increases as time goes by. To cancel this accumulating error, the central nervous system triggers catch-up saccades (de Brouwer et al., 2002; Fleuriet and Goffart, 2012; Daye et al., 2014). It is currently assumed that pursuit and saccadic commands are superimposed when subjects are tracking a moving target (de Brouwer et al., 2001, 2002; Fleuriet and Goffart, 2012; Daye et al., 2014, 2012; Kettner et al., 1996; Leung and Kettner, 1997). Therefore, it is common to assess the quality of the pursuit movement once the saccadic component has been removed (Kettner et al., 1996; Leung and Kettner, 1997; Daye et al., 2012). This assessment requires that saccades be detected accurately, and their onset and offset times be determined precisely. When the target velocity is constant, a constant velocity threshold can be used to detect the rapid increase of eye velocity, which is the traditional hallmark of saccades (Fleuriet and Goffart, 2012). Another possibility is to use an acceleration threshold which cancels the constant velocity term of linear pursuit (de Brouwer et al., 2002). Finally, Wyatt (1998) proposed using a threshold on jerk (third derivative of eye position) to detect saccade occurrence. Lesions of the central nervous system can give rise to post-saccadic drifts (Cannon and Robinson, 1987; Anastasio and Robinson, 1991; Optican et al., 1986; Robinson, 1974). In the case of post-saccadic drifts, as for catch-up saccades, a velocity threshold would be inadequate because of the polluting velocity component between saccades.

When the eye velocity is not constant (e.g. sinusoidal or exponential), successive derivatives do not cancel the smooth pursuit component and thus using a constant threshold becomes inadequate. Finally, it is now commonly accepted that the range of saccade amplitudes goes from 0.05° (microsaccades, e.g. Ratliff and Riggins, 1950; Hafed et al., 2009; Martinez-Conde et al., 2009) to 40° (exploratory saccades, e.g. Bahill et al., 1975) when the head is fixed. Therefore, a supplementary challenge for a saccade detection algorithm is its ability to detect saccades across a large range of amplitudes.

One could create a variable threshold that would take into account the target motion, as proposed to analyze the slow phase component of the vestibulo-ocular reflex (Barnes, 1982). This approach would need to reevaluate the threshold function for each protocol and each subject. Twenty years ago, Sauter et al. (1991) devised a saccade detection algorithm based on the innovations (model output prediction errors) of a Kalman filter on an autoregressive process (AR) that predicts eye velocity. More recently, Liston et al. (2013) proposed an algorithm based on a median filter and template matching of the saccadic velocity profile to detect saccades during pursuit.

This paper will present a new saccade detection algorithm based on the variance of the weights of a particle filter (an ensemble of Bayesian estimators). We will show that the method can detect saccades across a large range of amplitudes. The saccades can be intermixed (or not) with pursuit movements or eye drifts because the method cancels the baseline velocity (smooth pursuit or drift component) that usually pollutes the quality of saccade detection.

2. Materials and methods

The saccade detection algorithm presented in this paper is based on the basic expression of a particle filter (Doucet et al., 2001; Simon, 2006). Particle filters are powerful estimators for non-linear non-Gaussian systems. In their primal form, their algorithmic implementation is simple; a number of particles track a trajectory through a state-space representation of the (non-linear) dynamics of the system. Each particle receives a weight function of how well it approximates the trajectory. Then the position of all the particles is updated according to the weights and a metric (e.g. median, mean, etc.) is applied to the population to estimate the state of the system. Compared to other non-linear estimators (e.g. the extended Kalman Filter), one advantage of particle filters is that they do not rely on a linearization of the estimated non-linear system. However, this brute-force approach of the Bayesian estimator is done at a higher computational cost than other methods.

Eye movements trajectories are not linear, and there is not a set of dynamical equations that represent their trajectory in general. Therefore, we decided to use particle filters for their capacity to approximate a large set of non-linear dynamical systems despite their higher computational cost. The equations of the algorithm are presented in the next section without a demonstration of their mathematical properties, which can be found elsewhere (Doucet et al., 2001; Simon, 2006). The different analyses and the detection algorithm proposed in this paper were programmed using the python programming language. Implementations in python and Matlab (The Mathworks, Natick, MA) are available from the authors.

2.1. Particle filter algorithm

Our goal is to estimate the states of a system representing the eye movements using our measurements. A particle filter is an example of a general Bayesian state estimator, which we will explain first. Then we describe how the general Bayesian state estimator framework is approximated by a particle filtering.

2.1.1. Bayesian estimator

A non-linear system can be described by the set of discrete state-space equations:

\[ x_{k+1} = h_k(x_k, o_k) \]  
\[ y_k = g_k(x_k, c_k). \]

In these equations, \( k \) represents the time index, \( x_k \) is the state and \( y_k \) is the output of the system (measured variable). Both the state and the output contain a time-varying noise (\( o_k \) for the state and \( c_k \) for the output). Importantly, no assumption on the nature of the noises is required by the Bayesian estimator framework. However, their probability density function (pdf) should be known and the two noise sequences must be independent and white. The goal of the Bayesian estimator is to estimate the conditional probability density function (cpdf) of \( x_k \) given the measurements \( Y_k = y_0, y_1, \ldots, y_k \).

This Bayesian estimation problem can be solved recursively if one knows the initial state \( x_0 \), the initial output \( y_0 \), and the initial cpdf:

\[ p(x_0|Y_0) = p(x_0). \]  
\[ p(x_1|Y_1) = \int p(x_0|x_{k-1})p(x_k|Y_{k-1})dx_{k-1}. \]

In (3), \( Y_0 \) is defined as the set of no measurement. Once \( p(x_0|Y_1) \) is known, an estimate \( \hat{x}_1 \) of the true state \( x_1 \) can be computed using the mean, median or any other metric of the cpdf.

The estimation of the cpdf is done in two steps: prediction and update. The prediction step evaluates the prior cpdf of the system at time \( k \) using the Chapman–Kolmogorov equation:

\[ p(x_k|Y_{k-1}) = \int p(x_{k}|x_{k-1})p(x_{k-1}|Y_{k-1})dx_{k-1}. \]

Note that \( p(x_k|Y_{k-1}) \) in (4) is a simplification of \( p(x_k|(x_{k-1}, y_{k-1})) \) because \( x_k \) is entirely determined by the knowledge of \( x_{k-1} \) and \( o_{k-1} \). The first cpdf in the integrand of (4) can be computed from
the knowledge of \( h_k(. \) ) and the known pdf of \( \omega_k \). The second cpdf in the integrand of (4) is only known at the initial time step (see (3)).

Once the prior cpdf is computed from Eq. (4), one can extract the \textit{a posteriori} pdf (update step) using:

\[
p(x_i|Y_k) = \frac{p(y_k|x_i)p(x_i|Y_{k-1})}{\int p(y_k|x_i)p(x_i|Y_{k-1})dx_i}.
\]  

(5)

Starting from the initial estimation of the state cpdf \( p(x_0|Y_0) \), one can recursively compute first the prior cpdf (prediction step) and then the posterior cpdf (prediction) at each time step.

2.1.2. Particle filter

The particle filter is a numerical implementation of the general Bayesian state estimator framework presented in the previous section. The first step is to generate a set of \( N \) state estimates, called particles, from the (assumed to be known) initial pdf \( p(x_0) \). This set represents a discrete estimate of the continuous distribution \( p(x_0) \).

Then, each particle \( (x_i^j, 1 \leq i \leq N) \) is propagated to the next step through the state equation (1):

\[
x_{k+1}^i = h_k(x_k^i, \omega_k^i) \quad \forall \; i = 1, \ldots, N.
\]  

(6)

Each \( \omega_k^i \) in (6) is drawn from the known pdf of \( \omega_k \). Next, we use the knowledge of the measurement equation (2) to compute the conditional likelihood of the output of each particle \( p(y_k|x_i^j) \). This will allow us to compute a weight for each particle depending on how well it evaluates the current measurement. The likelihood \( L_k \) that the estimation \( \hat{y}_k^i \) using the particle value \( x_k^i \) represents the actual measurement \( y^* \) can be written using:

\[
\hat{y}_k^i = g_k(x_k^i, \zeta_k^i)
\]  

(7)

\[
L_k = P(\hat{y}_k^i = y^*)|x_k = x_k^i)
\]  

(8)

\[
= p(\zeta_k^i = g_k^{-1}(y^*, x_k^i))
\]  

(9)

If one assumes that \( \zeta_k \) in (2) acts linearly on the output of the approximated system, one can simplify Eqs. (7)-(9):

\[
\hat{y}_k^i = g_k(x_k^i) + \zeta_k^i
\]  

(10)

\[
L_k = p(\zeta_k^i = y^* - g_k(x_k^i)).
\]  

(11)

The actual probability \( p(\zeta_k^i = y^* - g_k(x_k^i)) \) could not be computed exactly but is directly proportional to the estimation error \( \epsilon_k^i \). If \( \zeta_k \) is drawn from a Gaussian distribution\(^1\) with a zero mean and a covariance matrix \( R \), we can write:

\[
\epsilon_k^i = y^* - g_k(x_k^i)
\]  

(12)

\[
L_k \propto \frac{1}{(2\pi)^{m/2}|R|^{0.5}} e^{-|\epsilon_k^i|^2/(2R)}
\]  

(13)

Because all the \( L_k \) are proportional to the real probability distribution and we try to evaluate how well one particle evaluates the current measurement, we can normalize the distribution of \( L_k \) and compute the relative probability likelihood (also called importance weight) \( \theta_i \) that each particle has to evaluate the current measurement as:

\[
\theta_i = \frac{L_i}{\sum_i L_i} \quad i = 1, \ldots, N.
\]  

(14)

Using (14), a weight is attributed to each particle.

In the final step, a new set of \( N \) particles must be drawn from the discrete distribution of \( \theta_i \). This step is called the particle resampling and corresponds to the prediction step of Eq. (5). Several methods exist to draw the new particles from the probability distribution generated by \( \theta_i \); we will use a method presented in Ristic et al. (2004). The method is represented in panel B of Fig. 1.

To generate the new set of particles, we compute the cumulative distribution \( \Gamma \) of the relative probability distribution (14):

\[
\Gamma(i) = \sum_{j=0}^{i} \theta_j.
\]  

(15)

Remember that each value of \( \theta_i \) represents how well the particle \( x_i \) can be used to evaluate the actual state of the system at time \( k \). For each particle, a random number \( u \) is generated from a uniform distribution on [0, 1]. Then, the updated value of the particle \( i \) will be the value of the particle \( j \) such that:

\[
\Gamma(j-1) < u < \Gamma(j).
\]  

(16)

In Fig. 1B, the particle that will be updated will take the value of particle number 2. At the end of the resampling step, a new distribution of particles is generated with a cpdf \( p(x_i|Y_k) \).

From the description of the particle filter algorithm, it results that the importance weights \( \theta_i \) are proportional to the ratio of the posterior to the prior cpdfs (Crisan and Doucet, 2002):

\[
\theta_i \propto \frac{p(x_i|Y_k)}{p(x_i|Y_{k-1})}.
\]  

(17)

This ratio (called the Radon–Nikodym derivative, see below) is the foundation of our detection algorithm.

---

\(^1\) A case with a signal-dependent noise is shown in Section 3.
Finally, as explained in Section 2.1.1, several possibilities exist to evaluate the state estimation from the population of particles. We use the median of the particles’ distributions. Using the median instead of the mean, we are less sensitive to outliers. Therefore, we can reduce the number of particles used to approximate the signal and decrease the computational effort of the method.

2.2. Saccade detection and filter reset

To detect saccades with a particle filter, the first step is to choose a state model for (1) and a measurement equation for (2). If not stated otherwise, we used:

\[ x_{k+1} = x_k + \omega_k \]
\[ y_k = x_k + \xi_k. \]

In (18) and (19), \( \omega_k \) is a Gaussian noise process with zero mean and variance \( \Omega \) and \( \xi \) is a Gaussian noise process with a zero mean and variance \( \xi \). In the model we use, each output (represented by \( y_k \)) corresponds also to a state variable (represented by \( x_k \)). Both \( \Omega \) and \( \xi \) are parameters of the detection algorithm. At each time step, we evaluate the variance of the importance weights distribution composed of \( \hat{\theta}_i \):

\[ \mu_k = \mathbb{E}[\hat{\theta}_i] = \frac{1}{N}, \]
\[ \Upsilon_k = \mathbb{E}[(\hat{\theta}_i - \mu_k)^2] \]

The ratio (17) is called the Radon–Nikodým derivative (RND). It is an image of the rate of change between two probability distributions. As shown in Eq. (23), larger RNDs lead to a larger variance of the importance weights (and thus larger \( \Upsilon_k \)). The demonstration of the actual computation of RND from the Bayes’ recursion equations and the importance sampling theory is out of the scope of this paper. Nevertheless, we provide an intuitive explanation of the behavior of this ratio in our usage. Because \( p(x_k|y_{k-1}) \) represents the updated position of the particles at instant \( k \) and \( p(x_k|y_{k-1}) \) represents the predicted position of the particles at instant \( k - 1 \), the ratio represents how far the particles moved away from the prediction during one time step, between \( k \) and \( k - 1 \). Therefore if the prediction is correct (no saccades), \( \Upsilon_k \) will be small but if the prediction fails (occurrence of a saccade) \( \Upsilon_k \) will be large. Finally, because the method evaluates when the signal deviates from a “no-saccade” condition, one could argue that it could be done with a linear low-pass filter. However, a linear dynamical system would either induce a delay (if the time constant is long) or would not be sensitive to fast changes (if the time constant is short) when tracking a
non-linear signal. One could tune the linear system independently for each trial but this would be as much work as a manual detection.

A critical point to account for with the proposed detection method is a degenerate situation in which the difference between \( p(x_k|Y_k) \) and \( p(x_k|Y_{k-1}) \) is too large. Then all the resampled particles will collapse to the same position (called the black hole of particle filtering) and the variance of \( p(x_k|Y_k) \) will be equal to zero leading to \( \Upsilon_k \) also equal to zero. Therefore, the algorithm must detect the samples when \( \Upsilon_k \) is larger than a threshold value but also when it is strictly equal to zero. If nothing is done when there is a particle filtering black hole, the trajectory of the estimated signal \( y^*_k \) can no longer be tracked. To avoid this situation, we reset the particle filter when \( \Upsilon_k \) is larger than a reset threshold \( \lambda \) or when it is equal to zero. This is done by computing a new set of particles around the current value of the estimated signal \( y^*_k \) with the initial distribution \( p(x_0) \).

Finally, because of the asymmetrical nature of saccades' velocity profile, called the saccadic skewness (Van Opstal and Van Gisbergen, 1987), we ran the filter once with increasing time samples (\( \Upsilon \) for time samples from 0 to \( k \)) and once with decreasing time samples (\( \hat{\Upsilon} \) for time samples from \( k \) to 0). The algorithm detects when both \( \Upsilon \) and \( \hat{\Upsilon} \) are above the threshold. Importantly, this step can be avoided if one wants to use the method on-line to change the target motion as a function of the subject’s behavior, as in saccadic adaptation (McLaughlin, 1967). A mathematical demonstration of the reason why the combination of the direct and the reverse order gives better results is outside the scope of this paper.

### 2.3. Grouping and repetitions

A final issue arises because of the noisy nature of \( \Upsilon_k \) that could generate either false positive or false negative detections. To prevent this problem, we bootstrap the particle filter \( m \) times and compute the median of the distributions \( \Upsilon \) and \( \hat{\Upsilon} \). Then, we detected the samples for which \( \Upsilon_k \) and \( \hat{\Upsilon} \) are larger than a threshold \( \psi \). Because of the stochastic nature of \( \Upsilon_k \) and \( \hat{\Upsilon} \), it is highly likely (this is visible in the figures in Section 3) that one of the two variances is larger than the threshold \( \psi \) during a small number of samples even if there is no saccade. On the contrary, the variances \( \Upsilon_k \) and \( \hat{\Upsilon} \) could be smaller than \( \psi \) for a few samples during a saccade. To avoid this problem, we grouped the values higher than the threshold that were distant by less than \( \sigma_k \) time samples. Finally, we remove groups that contain less than \( G \) points because they could not correspond to a saccade.

### 3. Results

In the following sections, the new algorithm will be applied to a variety of protocols used to test eye movements. Then we will present how the algorithm deals with the noise in the signal and we will conclude by demonstrating how a modification of the method’s parameters affects the particle filter behavior and thus the accuracy of the detection.
3.1. Microsaccades

Fig. 2 presents data acquired at 1 kHz while a monkey fixated a central target, which were kindly provided by Dr. Z. Hafed. Panel A represents a trial when the monkey fixated a central target. Panel B represents a trial in which the target was stabilized on the eye, with a little position error on the right to elicit stair-case saccades. Because of the two conditions, the trial in panel A contains only microsaccades while the trial in panel B combines microsaccades, macrosaccades and post-saccadic drifts between macrosaccades. The last row represents the time course of $\Upsilon$, $\hat{\Upsilon}$ and the threshold ($\psi = 0.0002$) used to detect the saccades.

In panel A, the range of saccade amplitudes goes from 0.09° (peak velocity: 4.33°/s) to 0.73° (peak velocity: 24.18°/s). In panel B, the range of saccade amplitudes goes from 0.34° (peak velocity: 5.97°/s) to 2.39° (peak velocity: 72.25°/s). Despite the very different characteristics of the saccadic eye movements between the two conditions, the same set of parameters was used for panels A and B (see legend of Fig. 2 for the set of parameters applied). First and second rows in panel B show that the eye drifts between large saccades. This is particularly clear when one compares the velocity traces between panels A and B; there is an obvious velocity component between large saccades in panel B that is not present in panel A. A key feature of the algorithm is that it ignores this residual velocity component. By comparing the last row in panels A and B, one can see that the drift component between saccades has vanished in panel B, making $\hat{\Upsilon}$ very similar for both trials.

3.2. Saccades during constant velocity pursuit

Fig. 3 represents binocular eye movements acquired with an Eyelink 2000 (SR Research Ltd., Canada) at 1 kHz. The data were not filtered. The subject had to look at a red dot moving with a step-ramp motion (represented by the red line in Fig. 3 on an LCD screen (refresh rate: 144 Hz). The target stayed at the center of the screen for 400 ms, then it jumped 2° to the left and moved to the right with a constant speed of 20°/s (Rashbass paradigm Rashbass, 1961). The green and the orange lines in the upper row of Fig. 3 represent respectively the time course of the left and right eye positions. The second row represents the evolution of the left ($\psi^*_L$, green line) and the right ($\psi^*_R$, orange line) eye velocities as well as their approximation by the particle filter (respectively $\hat{\psi}^*_L$, light blue line and $\hat{\psi}^*_R$, pink line) as a function of time. The last row represents the evolution of the detection signal of the left ($\Upsilon^*_L$, purple line) and the right ($\Upsilon^*_R$, blue line) as a function of time (for the sake of clarity, we do not show the reversed time sequences $\hat{\Upsilon}^*_L$ and $\hat{\Upsilon}^*_R$).

A saccade was detected when both $\Upsilon^*_L$ and $\Upsilon^*_R$ were above the threshold $\psi$ (red line in Fig. 3). It is clear that because the velocity is not constant during the whole trial, a constant velocity threshold would not permit a good detection of saccades during both the fixation and the pursuit parts. One could use the acceleration of the eye (a supplementary differentiation of the signal would cancel the constant term in the velocity profile), but it would increase the amount of noise in the baseline. As shown in the last row of Fig. 3, the baseline of the signals built using the particle filter ($\hat{\Upsilon}^*_L$) does not vary despite the velocity changes during the trial. Therefore, we
could use the same threshold for both the fixation and the pursuit parts.

3.3. Saccades during variable velocity pursuit

The previous sections showed how the new algorithm cancels the constant part of the velocity signal when a subject pursued a target moving at a constant velocity. Fig. 4 shows an experiment during which the subject had to pursue a target moving with a sinusoidal velocity (sine amplitude: 12°, sine frequency: 0.75 Hz). The same setup as in Section 3.2 was used to record the eye movements. The data were filtered with a moving average window (20 ms wide window). The third row in Fig. 4 shows that the baseline of \( \bar{\gamma} \) remains constant and is not modulated by the eye velocity signals. Therefore, as for the previous sections, a constant value for the detection threshold \( \psi \) can be used to detect saccades in the case of a non-linear target motion. Because no supplementary differentiation would suppress the modulations of target velocity as a function of time, our algorithm appears more efficient in this situation than a traditional velocity, acceleration or jerk threshold. Finally, the algorithm detected all the saccades in the trial despite the wide range of amplitudes (smallest amplitude: 0.20° with peak velocity: 19.22°/s. Largest amplitude: 6.64 with peak velocity: 180.5°/s).

3.4. Saccades during free viewing

Fig. 5 shows eye movement recordings when a subject was required to guess the name of a painting. Eye position was calibrated using a nine point pattern. The data were not filtered. Data in Fig. 5 are represented in the image reference frame (notice the pixel units in the figure). The trial lasted 20 s but for the sake of clarity only the first 4 s are presented in this figure. Upper row of panel A represents the time course of the horizontal (green line) and the vertical (orange line) eye position. Black lines represent saccades detected by the algorithm. The middle row of panel A represents the time course of the vectorial eye velocity \( \vec{v} = \sqrt{H^2 + V^2} \), green line and the approximation of the particle filter \( \hat{y} \), orange line. The last row of panel A represents the time course of \( \hat{\theta} \) and the threshold \( \hat{\psi} \) used to detect the saccades. Panel B in Fig. 5 represents the spatial position of the eye on the painting during the trial. Orange lines represent the fixation periods and green lines represent saccades.

3.5. Noise sensitivity

This section analyzes the sensitivity of the detection algorithm to the noise in the signal. To quantify the effect of the noise, we built a trial (representing an eye velocity trace) with:

\[
y^v(k) = 18\pi \cos(2\pi \cdot 0.75 \cdot k \cdot \Delta t) + 5 \cdot \sin(2\pi \cdot 7.5 \cdot k \cdot \Delta t). \tag{24}
\]

The duration of a trial was set to 1.33 s and the sample time was set to 1 ms. We generated 100 trials using (24) to create a block. Then we added a single triangular perturbation to each trial with a peak amplitude of 40°/s and a duration of 30 ms. The onset of the perturbation varied from trial to trial and spanned a range between 0.133 s and 1.066 s by steps of 9.5 ms. Finally we added noise to each trial with a variance \( \sigma_n^2 \) (noise signal, deg²/s²). We generated 16 blocks with different noise variances from \( \sigma_n^2 = 0 \) (no noise) to \( \sigma_n^2 = 3 \) by steps of 0.2. Panel A in Fig. 6 represents four trials picked from four different blocks. The thick black lines represent the saccade detected by the algorithm in the trials.

The parameters of the algorithm used for the different noise levels are presented in Table 1. They were set by hand and kept constant for all trials within a block. Then we computed the difference between the known onset (offset) of each perturbation and the onset (offset) detected by the algorithm. Fig. 6B represents the evolution of the mean error for the onset (thick green line) and the offset (thick blue line) as a function of \( \sigma_n^2 \). Thin lines represent the evolution of the standard error of the mean around the mean error as a function of \( \sigma_n^2 \). A positive error corresponds to an early detection and a negative error corresponds to a late detection.

For low levels of noise, the algorithm detects the offset \((0.68 \pm 0.05)\) ms too soon and the onset \((1.12 \pm 0.03)\) ms too late.
Table 1
Values of the different parameters used for the noise sensitivity analysis. The number of particles $N$ was set to 50. The number of repetitions $m$ was set to 5. The reset threshold $\lambda$ was set to 0.005.

<table>
<thead>
<tr>
<th>$\sigma_s^2$</th>
<th>$\Omega$</th>
<th>$\xi$</th>
<th>$\psi$</th>
<th>$G_d$</th>
<th>$G_m$</th>
<th>$\sigma_s^2$</th>
<th>$\Omega$</th>
<th>$\xi$</th>
<th>$\psi$</th>
<th>$G_d$</th>
<th>$G_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>1.5</td>
<td>2e−4</td>
<td>10</td>
<td>10</td>
<td>1.6</td>
<td>2.8</td>
<td>12.0</td>
<td>1.5e−4</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>0.2</td>
<td>2.1</td>
<td>2.7</td>
<td>1.5e−4</td>
<td>10</td>
<td>10</td>
<td>1.8</td>
<td>2.9</td>
<td>13.5</td>
<td>2e−4</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>0.4</td>
<td>2.2</td>
<td>3.9</td>
<td>1.5e−4</td>
<td>10</td>
<td>10</td>
<td>2.2</td>
<td>3.2</td>
<td>15</td>
<td>2e−4</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>0.6</td>
<td>2.3</td>
<td>5.1</td>
<td>2e−4</td>
<td>10</td>
<td>10</td>
<td>2.6</td>
<td>3.6</td>
<td>35</td>
<td>1e−4</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>0.8</td>
<td>2.4</td>
<td>6.3</td>
<td>2e−4</td>
<td>10</td>
<td>10</td>
<td>2.8</td>
<td>3.8</td>
<td>51.67</td>
<td>1e−4</td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td>1.0</td>
<td>2.6</td>
<td>9</td>
<td>2e−4</td>
<td>10</td>
<td>10</td>
<td>3</td>
<td>4</td>
<td>60</td>
<td>1e−4</td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td>1.2</td>
<td>2.7</td>
<td>10.5</td>
<td>2e−4</td>
<td>10</td>
<td>10</td>
<td>4</td>
<td>6</td>
<td>8e−5</td>
<td>60</td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td>1.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For the highest level of noise ($\sigma_s^2 = 3$), the algorithm detects the offset (3.36 ± 0.43) ms too late and the onset (3.35 ± 0.57) ms too late. Within this range of noise variance, the evolution of the detection error is close to linear. Finally, a key point is the spread of the error. Fig. 6 shows that the standard error of the mean of the error is small throughout the range of variance, meaning that the behavior of the algorithm remains consistent despite the level of noise. We did not test levels of noise higher than $\sigma_s^2 = 3$ because it

![Fig. 6. Onset/offset detection sensitivity to signal noise. Panel A represents four trials with different levels of noise. The dataset used in this figure was simulated. The thin black lines represent the simulated eye velocity. The thick black lines represent the saccade detected by the algorithm. Panel B represents the evolution of the onset/offset detection error as a function of the variance of the noise added to the signal. The thick blue (green) line represents the mean temporal error on the offset (onset) detection as a function of the signal noise variance. Thin colored lines represent the evolution of the standard error of the mean around the mean as a function of the signal noise ratio. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)](Image 166x146 to 502x594)
is clear that the signal should be cleaned first with a filter before using the proposed algorithm in these conditions.

3.6. Signal-dependent noise

Using a video eyetracker, the noise of the camera is approximately equal over the measurement range but the camera response is a function of the eye eccentricity. Therefore, the noise increases with increasing eye eccentricity (SR Research, personal communication). Because the particle filter can be used to track dynamical states with non-Gaussian signal-dependent noise, we generated several 1s trials using Eq. (24) to which we added a signal-dependent noise \( \nu(y^*) \) defined as:

\[
\nu(y^*) = N(0, 5 \cdot (1.1 - |\cos(0.0129 \cdot y^*)|)).
\]

\( N(\mu, \sigma^2) \) in Eq. (25) represents a Gaussian distribution with mean \( \mu \) and variance \( \sigma^2 \). Therefore, the noise variance is a function of the eccentricity \( y^*(k) \) \( \sigma^2(y^* = 0) = 0.5, \sigma^2(y^* = 60) = 5.5 \). Finally, we added the same signal-dependent measurement noise instead of a Gaussian noise in Eq. (13):

\[
\phi(y^*) = 1 + 1.5 \cdot |\sin(0.0129 \cdot y^*)|
\]

\[
\chi(y^*) = \frac{1}{\int_{-2\phi(y^*)}^{2\phi(y^*)} \Lambda_1 dy^*}
\]

\[
\Lambda_1 \propto \begin{cases} 
0 & \text{if } \epsilon^i \leq -2\phi(y^*) \\
(2 + \epsilon^i)\chi(y^*) & \text{if } -2\phi(y^*) < \epsilon^i < -\phi(y^*) \\
\chi(y^*) & \text{if } -\phi(y^*) \leq \epsilon^i \leq \phi(y^*) \\
(2 - \epsilon^i)\chi(y^*) & \text{if } \phi(y^*) < \epsilon^i < 2\phi(y^*) \\
0 & \text{if } \epsilon^i \geq 2\phi(y^*)
\end{cases}
\]

The shape of the probability distribution \( \Lambda_1 \) is shown in Fig. 7. Panel A. Eq. (26) represents the dependency of the noise on the signal while Eq. (27) computes the normalization factor needed to ensure that Eq. (28) represents a probability distribution. The shape of the probability distribution of the measurement error has been chosen because we postulated that when the signal is close to the “real” eye position, there is a range of measurement with equal probabilities. All the parameters of the filter, except the variance of the measurement noise \( \xi \), are given in the legend of Fig. 7. As in Section 3.5, we computed the error between the known offset time of each perturbation and the detected onset (offset) time for each perturbation. The detection algorithm detected the onset \( 1.6 \pm 0.7 \) ms too soon and the offset \( 2.3 \pm 0.5 \) ms too late. These results demonstrate that the average error is in the same range as the errors measured in Section 3.5 with a Gaussian noise. Therefore, our detection algorithm, based on a particle filter, can detect saccades in signals polluted with non-Gaussian signal-dependent noises.

3.7. Tuning of the detection algorithm parameters

This section provides the user with some intuition about the consequences of a modification of the different parameters on the detection algorithm. To tune the behavior of the detection algorithm, the user can change the variance \( \Omega \) of the state process (18), the variance \( \xi \) of the measurement Eq. (19), the number \( N \) of particles, the number \( m \) of repetitions, the reset threshold \( \lambda \) and the detection threshold \( \psi \). The detection threshold \( \psi \) is the most straightforward parameter to tune, it must be chosen as small as possible while avoiding the noise in \( \Upsilon \).

\[ y^k = \sin(2\pi \cdot 1.1 \cdot k \cdot \Delta t) + 0.05 \cdot \sin(2\pi \cdot 17.5 \cdot k \cdot \Delta t) + \varphi(k). \]

In (29), the sample time \( \Delta t \) was set to 1 ms, the duration of the simulation was set to 0.5 s and \( \varphi(k) \) was drawn from a Gaussian distribution with a mean 0 and a variance of 0.01. Between 0.15 and 0.17 s, we added a triangular perturbation with a peak amplitude of 0.5. To quantify how the parameters affect the quality of the detection, we computed a ratio \( G(\Upsilon) \) between the median value of \( \Upsilon (\hat{\Upsilon}) \) during the perturbation and the standard deviation of \( \Upsilon (\hat{\Upsilon}) \) without the perturbation. This index expresses how distinguishable is the perturbation with respect to the overall behavior; the higher the index, the easier the detection. Because the method detects when both \( \Upsilon \) and \( \hat{\Upsilon} \) are above the threshold, both \( C \) and \( \hat{C} \) must be large to guarantee the detection quality. We also computed the mean
squared error (MSE) of the estimation of the tracked trajectories for the conditions tested.

Panel A shows that independently of the number of particles in the filter, the trajectory is correctly approximated by the filter (MSE is very similar between the conditions as a function of the number of particles in Table 2). However, $\Omega$ in Table 2 shows that the detectability of the event increases with increasing $N$ (this is predictable because the approximation of the continuous pdfs by the discrete set of particles improves when $N$ increases). For the sake of the demonstration, we kept a fixed threshold for all the examples to stress that an event could remain undetected because of a poor parameter choice. Therefore, to detect the event in the rightmost (leftmost) column of panel A, the threshold $\psi$ should be decreased (increased). Finally, increasing $N$ increases the computational effort to run the particle filter and is theoretically not a guarantee of accurate tracking. Therefore, a trade-off must be found between the number of particles and the time needed to extract the saccades.

Panel B in Fig. 8 shows that the tracking quality of the target depends strongly on the choice of $\Omega$. A small $\Omega$ will prevent the filter from evaluating fast changing signals. This can be seen in the first column of panel B. The green line (forward filter) is delayed with respect to the approximated signal (orange line). The same observation is true for the reverse filter if one looks at the time in the reverse direction (from 0.5 to 0 s). When one looks at the time course of $\hat{\gamma}$ and $\hat{\gamma}$, one can see that there is a strong modulation of these signals outside the perturbation. A different pathological behavior arises when too much freedom is given to the filter. This is the case in the rightmost column of panel B. In this situation, the signal is perfectly tracked but there is no modulation of $\hat{\gamma}$ or $\hat{\gamma}$ linked to the perturbation (this is a sign of over-fitting). This is summarized in Table 2. $\Omega$ is poor for small values of $\Omega$ as well as for large values of $\Omega$.

The last row of Fig. 8 shows a similar story to the one of panel B for $\xi$. However the amplitude of the variations of $\hat{\gamma}$ and $\hat{\gamma}$ decreases with increasing $\xi$. The quality of the tracking of the trajectory is worse for large values of $\xi$ than for small $\xi$ (larger $\xi$ in Table 2 correspond to larger MSE).

4. Discussion

We proposed a new algorithm to detect saccades based on the weights’ distribution of a particle filter. We presented data that demonstrate the ability of the algorithm to detect saccadic movements within a large range of amplitudes (from microsaccades to
Table 2  
Ratio $G$ and mean squared error (MSE) of the filter tracking for the different conditions shown in Fig. 8 for the forward filter ($\hat{Y}$) and the reversed filter ($\hat{Y}'$). 

<table>
<thead>
<tr>
<th></th>
<th>N = 5</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\hat{Y}$</td>
<td></td>
<td>$\hat{Y}'$</td>
<td></td>
<td>$\hat{Y}$</td>
</tr>
<tr>
<td>N = 50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>2.55</td>
<td>1.96e−4</td>
<td>1.50e−4</td>
<td>10.79</td>
<td>1.39e−4</td>
</tr>
<tr>
<td>$\Omega = 2e−5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{Y}$</td>
<td></td>
<td></td>
<td>$\hat{Y}$</td>
<td></td>
<td>$\hat{Y}$</td>
</tr>
<tr>
<td>N = 500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>3.00</td>
<td>0.036</td>
<td>0.036</td>
<td>10.67</td>
<td>1.60e−4</td>
</tr>
<tr>
<td>$\xi = 3e−4$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{Y}$</td>
<td></td>
<td></td>
<td>$\hat{Y}$</td>
<td></td>
<td>$\hat{Y}$</td>
</tr>
<tr>
<td>N = 500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>5.50</td>
<td>1.19e−5</td>
<td>1.14e−5</td>
<td>11.48</td>
<td>1.54e−4</td>
</tr>
</tbody>
</table>

exploratory saccades) and for a large set of protocols (from free viewing to variable velocity pursuit). We also showed that the algorithm parameters could be easily tuned to detect saccades in unfiltered signals. This is a key advantage of the proposed algorithm because it makes it suitable for on-line detection of saccades. On-line detection can be done if one uses only the forward computation of $\hat{Y}$. It must be noted that the use of both the forward and reversed time order particle filters is done to avoid a bias in the detection of saccade offset linked to the skewness of the saccade and to be more robust to the noise in the signal. Because on-line detection is traditionally used to change the paradigm behavior during the task, most of the time only the onset detection is critical (e.g. for saccadic adaptation McLaughlin, 1967).

A major difference between our method and the one proposed by others to detect saccades during pursuit (Sauter et al., 1991; Liston et al., 2013) is that it uses a very generic and simple model to represent eye velocity (see Eq. (18)). In its current derivation, the Kalman filter method proposed by Sauter et al. (1991) first estimates an autoregressive model on the eye velocity traces that will then be used by the Kalman filter. Obviously, this step cannot be done on-line. The same remark applies to the median filter method proposed by Liston et al. (2013). To avoid estimation biases, the moving window of the median filter must be centered around the current time step (meaning that you need as many time samples before and after the current time sample to avoid biases with the median filter). Finally, our method does not use a matching template to detect saccades as in (Liston et al., 2013). The use of a template makes sense when one is looking at (stereotyped) saccades of a healthy subject with the head fixed, but detection of saccades that do not comply with the chosen template will not be adequate. This is the case for patient recordings (e.g. Zee et al., 1976; Daye et al., 2013) or when the head is free to move (e.g. Gandhi, 2011; Daye et al., 2014; Freedman and Sparks, 1997). In these conditions, the saccade velocity profiles can be very different from the stereotyped ones in healthy, head-fixed, subjects. Thus a matching template approach would give inaccurate saccade detection in these situations.

Although the current algorithm has several advantages compared to other methods, it has a high computational burden. When targets are stationary and subjects have their heads held, a simple velocity threshold is sufficient to detect saccades. However, when there are eye movements between saccades, or when the range of saccadic amplitudes is large (from microsaccades to exploratory saccades), a simple threshold would not be sufficient to detect saccades accurately and reliably. In these conditions, the computational burden of our new method is offset by its robustness.

References

Fleuret J, Goffart L. Saccadic interpolation of a moving visual target after a saccade- 


